1. Design and implement a class named InstanceCounter to track and count the number of instances created from this class.

class InstanceCounter{

     static int count;

     void increment\_counter(){

         InstanceCounter.count++;

     }

    InstanceCounter(){

        this.increment\_counter();

    }

}

public class Q1 {

    public static void main(String[] args) {

        InstanceCounter c = new InstanceCounter();

        InstanceCounter c2 = new InstanceCounter();

        InstanceCounter c3= new InstanceCounter();

        InstanceCounter c4 = new InstanceCounter();

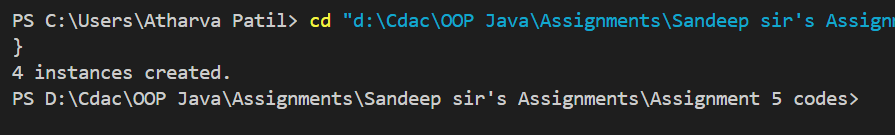
        int instanceCreated = InstanceCounter.count;

        System.out.println(instanceCreated + " instances created.");

    }

}

Output:



1. Design and implement a class named Logger to manage logging messages for an application. The class should be implemented as a singleton to ensure that only one instance of the Logger exists throughout the application

Code:

class Logger{

    private String log;

    private static  Logger logRef;

    private Logger(String log){

        this.log = log;

    }

     static Logger getInstance(String log){

        if (logRef == null){

            logRef = new Logger(log);

        }

        return Logger.logRef;

    }

    String getLog(){

       return this.log;

    }

    void clearLog(){

        this.log = null;

    }

}

public class Q2 {

    public static void main(String[] args) {

        Logger l = Logger.getInstance("Hello");

        System.out.println(l.getLog());

        l.clearLog();

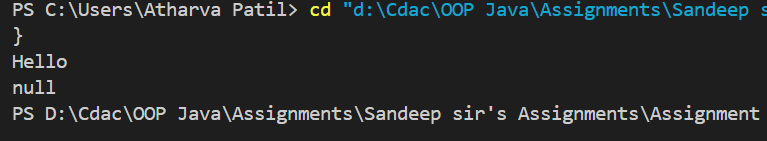
        Logger l2 = Logger.getInstance("Hello world");

        System.out.println(l2.getLog());

    }

}

Output:



1. Design and implement a class named Employee to manage employee data for a company. The class should include fields to keep track of the total number of employees and the total salary expense, as well as individual employee details such as their ID, name, and salary.

import java.util.PriorityQueue;

class Employee{

    private String Name;

    private int EmpId;

    private double Salary;

    static int EmpCount;

    static double TotalSalaryExpense;

    Employee(String name,int id, double salary){

        this.Name = name;

        this.EmpId = id;

        this.Salary = salary;

        EmpCount++;

        TotalSalaryExpense+=this.Salary;

    }

    int getTotalEmployees(){

        return Employee.EmpCount;

    }

    void calculateTotalSalaryExpense(double amount){

        Employee.TotalSalaryExpense += amount;

    }

    void applyRaise(double percent){

        double amount = (this.Salary/100)\*percent;

        this.Salary = this.Salary + amount;

        calculateTotalSalaryExpense(amount);

    }

    void getEmployeeDetails(){

        System.out.println("Name: "+this.Name);

        System.out.println("ID: "+this.EmpId);

        System.out.println("Salary: "+this.Salary);

    }

}

public class Q3 {

    public static void main(String[] args) {

        Employee e1 = new Employee("Atharva",17,45000.24);

        Employee e2 = new Employee("Aditya",117,55000.24);

        Employee e3 = new Employee("Anchal",174,45350.30);

        Employee e4 = new Employee("Amey",317,49500);

        Employee e5 = new Employee("govind",147,60200.40);

        System.out.println("Before salary update ");

        System.out.println();

        e2.getEmployeeDetails();

        System.out.println("Total Employee count: "+Employee.EmpCount);

        System.out.println("Total Salary Expenses: "+Employee.TotalSalaryExpense);

        e2.applyRaise(30);

        System.out.println("After salary update ");

        System.out.println();

        e2.getEmployeeDetails();

        System.out.println("Total Employee count: "+Employee.EmpCount);

        System.out.println("Total Salary Expenses: "+Employee.TotalSalaryExpense);

    }

}

Output:

